



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/599,299	06/21/2000	Sarita M. James	MSI-559US	6995

22801 7590 10/28/2003

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER

GROSS, KENNETH A

ART UNIT PAPER NUMBER

2122

DATE MAILED: 10/28/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

my

Office Action Summary

Application No.

09/599,299

Applicant(s)

JAMES ET AL.

Examiner

Kenneth A Gross

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☐ Responsive to communication(s) filed on ____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-96 is/are pending in the application.
- 4a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) ____ is/are allowed.
- 6) ☒ Claim(s) 1-96 is/are rejected.
- 7) ☐ Claim(s) ____ is/are objected to.
- 8) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on ____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on ____ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) ____.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). ____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

Claim Objections

1. Claims 1 and 81 are objected to because of the following informalities: The term “said describing defining” in Claim 1 should be “said describing includes defining” or some similar form. The term “statistics are ascertaining” in Claim 81 should be “statistics are [ascertaining] ascertained”.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 63, 93, and 94 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, Claim 63 states the term “old package directory”, which is unclear. The term “old” is indefinite, and can represent a directory that has existed for a long time, a directory that has existed before a new directory, a directory that stores old packages, or a directory that used to store packages. Claim 93 states the term “no file group information is provided” and also states “files of different types”. It would seem that if the files all have different file types, then this would represent file group information. Claim 94 states the term “average order in which the files were downloaded”. However, different files are always

Art Unit: 2122

being downloaded. Does this Claim indicate that files are sorted by their average priorities when they are currently slated for download, and not their average file download order?

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claim 82 is rejected under 35 U.S.C. 102(e) as being anticipated by Rowley (U.S. Patent Number 5,999,740).

In regard to Claim 82, Rowley teaches: (a) receiving information pertaining to extension directories that contain files that comprise a software extension to be incorporated on a client (Column 4, lines 60-61); (b) receiving information pertaining to file groups or load dependencies (Column 4, lines 60-61); (c) receiving information pertaining to file usage statistics. Rowley teaches inputting a flag indicating the permissions of the file, which is a file usage statistic (Figure 8); (d) generating a package manifest based on the received information (Column 4, lines 57-60).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1-5, 8-20, 37-39, 42-46, 55, 57, 61, 62, and 71-76 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Iannucci et al. (U.S. Patent Number 6,219,698).

In regard to Claim 1, Rowley teaches: (a) describing one or more software extensions configured for incorporation on a client, where describing includes defining one or more manifests containing a list of files comprising an extension (Column 2, lines 24-30); (b) delivering the one or more manifests to the client via the network, the one or more manifests being configured for use in downloading the software extensions via the network (Column 1, lines 40-50). Rowley does not teach that the software extensions are described using a hierarchical language. Iannucci, however, does teach a manifest file in the form of a web page, which is sent to the client for use in downloading a software extension (Column 5, lines 47-51). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to describe one or more software extensions in one or more manifests containing a list of files comprising an extension, and deliver the one or more manifests to the client via the network, the one or more manifests being configured for use in downloading the software extensions via the network, as taught by Rowley, where the software extensions are described using a hierarchical language, since this allows the manifest file to take the form of a web page.

In regard to Claim 2, Rowley teaches that the manifest is used to determine which files are to be downloaded to a client, and hence assists in organizing the delivery of individual files that the client lacks (Column 1, lines 40-50).

In regard to Claim 3, Rowley teaches comparing individual files in the manifest with the client system, and hence the manifest assists in validating the files to see which files need to be downloaded to the client (Column 1, lines 40-50).

In regard to Claim 4, Rowley teaches updating an application by downloading individual files, where individual files in the client system are replaced with updated files (Column 7, lines 41-44).

In regard to Claim 5, Rowley does not specifically teach that the manifests describe individual file locations. Iannucci, however, does teach a manifest with a link to the software, the link being the address of the current software (Column 5, lines 47-51). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the manifest describes individual file locations, as taught by Innaucci, since this allows the client to access the individual files by their location.

In regard to Claim 8, Rowley does not teach that the manifests are defined in a tag-based language. Innaucci, however, does teach a web page based manifest, which is obviously written in a tag-based language (Column 5, lines 47-51). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the manifest is written in a tag-based language, since this allows the manifest file to take the form of a web page.

In regard to Claim 9, Rowley does not teach that the manifests are defined in XML. However, XML is a well-known and popular language for creating web pages. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the manifest is written in XML, since this allows the manifest to use the features of the XML language.

In regard to Claim 10, Rowley does not teach that the network is the Internet. Innaucci, however, does teach distributing files over the Internet (Column 1, lines 10). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the network is the Internet, since this allows for a greater number of users to access software upgrades.

In regard to Claims 11 and 12, it would be obvious to include the method of Claim 1 on a media and a computer system, the method of Claim 1 describes software and networks, which would inherently involve the need for the method to be implemented on a computer system, and stored on a computer media.

In regard to Claim 13, Rowley teaches: (a) describing one or more software extensions configured for incorporation on a client, where describing includes defining one or more manifests containing a list of files comprising an extension (Column 2, lines 24-30). Rowley teaches that the manifest is used to determine which files are to be downloaded to a client, and hence assists in organizing the delivery of individual files that the client lacks (Column 1, lines 40-50). Rowley teaches comparing individual files in the manifest with the client system, and hence the manifest assists in validating the files to see which files need to be downloaded to the client (Column 1, lines 40-50). Rowley teaches updating an application by downloading

individual files, where individual files in the client system are replaced with updated files (Column 7, lines 41-44); (b) delivering the one or more manifests to the client via the network, the one or more manifests being configured for use in downloading the software extensions via the network (Column 1, lines 40-50). Rowley does not teach that that software extensions are defined using XML. Innaucci, however, does teach a manifest file in the form of a web page, which is sent to the client for use in downloading a software extension (Column 5, lines 47-51). XML is a well-known and popular language for creating web pages. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to describe one or more software extensions in one or more manifests containing a list of files comprising an extension, and deliver the one or more manifests to the client via the network, the one or more manifests being configured for use in downloading the software extensions via the network, validating individual files listed in the manifest, and updating individual files in the manifest, as taught by Rowley, where the software extensions are described using XML, since this allows the manifest file to take the form of a web page with XML features.

Claim 14 corresponds directly with Claim 13, and is rejected for the same reasons as Claim 13.

In regard to Claim 15, Rowley teaches updating a computer client with a software upgrade (Figure 1, items 110 and 112).

In regard to Claim 16, the examiner takes official notice that reading a file is usually done in sequential order, and it would be obvious to perform checks in files in a manifest in sequential order as the file names are being read from the manifest file.

In regard to Claim 17, Rowley teaches comparing individual files in the manifest with the client system, and hence the manifest assists in validating the files to see which files need to be downloaded to the client (Column 1, lines 40-50).

In regard to Claim 18, Rowley teaches updating an application by downloading individual files, where individual files in the client system are replaced with updated files (Column 7, lines 41-44).

In regard to Claims 19 and 20, it would be obvious to include the method of Claim 14 on a media and a computer system, the method of Claim 14 describes software and networks, which would inherently involve the need for the method to be implemented on a computer system, and stored on a computer media.

In regard to Claim 37, Rowley teaches: (a) describing one or more software extensions configured for incorporation on a client, where describing includes defining one or more manifest files containing a list of files comprising an extension (Column 2, lines 24-30). Rowley does not teach that the manifest files are written in XML nor does he teach that the XML files are stored in a Web-accessible location. Innaucci, however, does teach a manifest file in the form of a web page, which is sent to the client for use in downloading a software extension (Column 5, lines 47-51). XML is a well-known and popular language for creating web pages. Furthermore, Innaucci teaches storing the manifest files in a Web-accessible location (Column 1, lines 30-37). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to describe one or more software extensions configured for incorporation on a client, where describing includes defining one or more manifest files containing a list of files comprising an extension, as taught by Rowley, where the manifest file is written in XML and

Art Unit: 2122

stored in a web-accessible location, as taught by Innaucci, since this allows for a greater number of users to access software upgrades and allows the manifest file to take the form of a web page with XML features.

In regard to Claim 38, Rowley does not teach storing extension files in a Web-accessible location. Innaucci, however, does teach storing the extension files in a Web-accessible location (Column 1, lines 30-37). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the extension files are stored in a web-accessible location, as taught by Innaucci, since this allows for a greater number of users to access software upgrades.

In regard to Claim 39, Rowley teaches providing a manifest file and associated extension files to a client (Column 1, lines 40-50). Rowley does not teach storing extension files in a Web-accessible location. Innaucci, however, does teach storing the extension files in a Web-accessible location (Column 1, lines 30-37). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the extension files are stored in a web-accessible location, as taught by Innaucci, since this allows for a greater number of users to access software upgrades.

In regard to Claim 42, Rowley teaches one or more file groups associated with files (Column 2, lines 25-28) that determine when files are downloaded to the client (Column 2, lines 39-40).

In regard to Claim 43, Rowley teaches that the file groups determine where files are stored on the client (Column 2, lines 34-35).

In regard to Claim 44, Rowley teaches that the file groups determine how files are packaged (Column 3, lines 49-50).

Claim 45 corresponds directly with Claims 43 and 44, and is rejected for the same reasons as Claim 43 and 44.

In regard to Claim 46, it would be obvious to include the method of Claim 37 on a media since the method of Claim 14 describes software and networks, which would inherently involve the need for the method to be implemented and stored on a computer media.

In regard to Claim 55, Rowley teaches a manifest file, which stores a list of files utilized in a software extension (Column 2, lines 24-30), and further teaches one or more file groups associated with files (Column 2, lines 25-28). Rowley does not teach that the individual files and file groups comprise tags indicate the individual files and file groups. Iannucci, however, does teach a manifest file in the form of a web page, which is sent to the client for use in downloading a software extension (Column 5, lines 47-51). Typically, web pages are designed using HTML, a tag based language. Using tags to separate fields in a web page is an inherent aspect of HTML. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to build a data structure which stores a list of files utilized in a software extension and further teaches one or more file groups associated with files, as taught by Rowley, where the files and file groups are indicated by tags, which is an inherent aspect of HTML, which is obviously used to construct a web page manifest, as taught by Iannucci, since this allows the manifest file to take the form of a web page.

In regard to Claim 57, Rowley teaches storing information on file dependencies in the data structure (Column 2, lines 55-58).

In regard to Claim 61, the examiner takes official notice that XML is a well-known and popular language for creating web pages.

In regard to Claim 62, Rowley teaches a manifest file, which stores a list of files utilized in a software extension (Column 2, lines 24-30) to be downloaded to the client, and the address of the target directory in which the package is downloaded to the client (Column 2, lines 34-35). Rowley does not teach a URL of a package manifest on a code server. Iannucci, however, does teach a manifest file in the form of a web page with a link to a software upgrade on a server (Column 5, lines 47-51). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build a data structure which stores a list of files utilized in a software extension and the address of the target directory in which the package is downloaded to the client, as taught by Rowley, where the data structure further contains teach a URL of a package manifest on a code server, as taught by Iannucci, since this allows the client to access the address of the package it wishes to download.

In regard to Claim 71, Rowley teaches: (a) receiving one or more input parameters pertaining to a package manifest that is to describe a software extension that is configured to extend a software application on a client. Figure 8 shows a package manifest editor that accepts parameters to create a manifest; (b) generating a package manifest that describes the extension (Column 4, lines 57-60). Rowley does not teach that the manifest is described using a hierarchical language. Iannucci, however, does teach a manifest file in the form of a web page, which is sent to the client for use in downloading a software extension (Column 5, lines 47-51). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to receive one or more input parameters pertaining to a package manifest that is to

Art Unit: 2122

describe a software extension that is configured to extend a software application on a client and generate a package manifest that describes the extension, as taught by Rowley, where the manifest is described using a hierarchical language, as taught by Iannucci, since this allows the manifest to take the form of a web page.

In regard to Claim 72, Claim 72 corresponds directly with Claim 8, and is rejected for the same reasons as Claim 8.

In regard to Claim 73, Claim 73 corresponds directly with Claim 9, and is rejected for the same reasons as Claim 9.

In regard to Claim 74, Rowley teaches inputting a directory containing files to be described in the manifest (Column 4, lines 60-61).

In regard to Claim 75, Rowley teaches inputting a directory, which is a file group (Column 4, lines 60-61).

In regard to Claim 76, Rowley teaches inputting a flag indicating the permissions of the file, which is a file usage statistic (Figure 8).

8. Claims 6, 40, 41, and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Iannucci et al. (U.S. Patent Number 6,219,698) and further in view of Swank (U.S. Patent Number 4,641,274).

In regard to Claim 6, Rowley and Iannucci teach the method of Claim 1, but do not teach that the manifests contain individual hashes for one or more listed files. Swank, however, does teach storing a file hash for an individual file to be updated on a computer system (Column 3, lines 45-51). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the manifests contain individual hashes

for one or more listed files, since this allows for better security and integrity when updating computer files on a computer system. Claim 40 corresponds directly with Claim 6, and is rejected for the same reasons as Claim 6.

In regard to Claim 41, Rowley and Iannucci teach the method of Claim 37, but do not teach that individual manifests can contain one or more file hashes used for versioning. Swank, however, does teach storing a file hash for an individual file to be updated on a computer system (Column 3, lines 45-51). A hash can identify individual files, and hence different versions of the same file would contain different hashes. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where individual manifests can contain one or more file hashes used for versioning, as taught by Swank, since this allows different versions of the same file to be uniquely identified.

In regard to Claim 56, Rowley and Iannucci teach the method of Claim 55, but do not teach but do not teach that the data structure contains individual hashes for one or more listed files. Swank, however, does teach storing a file hash for an individual file to be updated on a computer system (Column 3, lines 45-51). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to construct the data structure of Claim 55, where the manifests contain individual hashes for one or more listed files, since this allows for better security and integrity when updating computer files on a computer system.

9. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Iannucci et al. (U.S. Patent Number 6,219,698) and further in view of Miller et al. (U.S. Patent Number 5,195,183).

In regard to Claim 7, Rowley and Iannucci teach the method of Claim 1, but do not teach that the one or more manifests contain download directives for downloading the listed files. Miller, however, does teach transmitting download directives for the purposes of downloading files (Column 22, lines 31-37). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the one or more manifests contain download directives for downloading the listed files, since this allows for the automatic transmission of a file or files without user intervention.

10. Claims 21-24, 27, 28, 32, 36, 47, 49-51, 53, and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Swank (U.S. Patent Number 4,641,274).

In regard to Claim 21, Rowley teaches a manifest file which stores a list of files utilized in a software extension (Column 2, lines 24-30), and further teaches one or more file groups associated with files (Column 2, lines 25-28), the file groups determining what particular files of the extension get downloaded to the client. Rowley does not teach that the manifests contain individual hashes for listed files. Swank, however, does teach storing a file hash for an individual file to be updated on a computer system (Column 3, lines 45-51). Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to design a data structure, including stores a list of files utilized in a software extension, one or more file groups associated with files in the list, as taught by Rowley, and the manifests contain individual hashes for listed files, as taught by Swank, since this allows for better security and integrity when updating computer files on a computer system.

In regard to claim 22, Rowley teaches determining where files are stored on the client (Column 2, lines 34-35).

In regard to Claim 23, Rowley teaches that the file groups determine how files are packaged (Column 3, lines 49-50).

In regard to Claim 24, Claim 24 corresponds directly with Claims 22 and 23 and are rejected for the same reasons as Claims 22 and 23.

In regard to Claim 27, Rowley teaches installing updates in a computer system by recognizing the updates that the user needs. Hence, the updates that are recommended by Rowley are always required for the first time by a user (Figure 9).

In regard to Claim 28, Rowley teaches updates that are only downloaded when the user selects the updates, hence demands that the update be downloaded (Figure 9). The download can only take place when a user is online, since the files need to be downloaded via a client/server connection (Figure 1, item 103).

In regard to Claim 32, Rowley teaches storing the size of the file on the manifest (Column 2, lines 38).

In regard to Claim 36, Rowley does not teach that the data structure is defined in XML. However the examiner takes official notice that XML is a well-known and popular language for creating web pages, and it would be obvious to embody the data structure as an XML file, since this would make the manifest accessible as a web page.

In regard to Claim 47, Rowley teaches receiving a package manifest containing a list of multiple files that comprise a software extension (Column 1, lines 40-43) and receiving the multiple files that are described in the package manifest (Column 1, lines 45-50). Rowley does

Art Unit: 2122

not teach that the list contains a hash for one or more of the files, creating a hash for the one or more received files, and comparing the created hash with the corresponding file hashes contained in the manifest to ascertain security of the file. Swank, however, does teach storing a file hash for an individual file to be updated on a computer system, creating an updated hash of a received file, and comparing the hashes to determine the integrity of the file (Column 3, lines 45-51).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to receive a package manifest containing a list of multiple files that comprise a software extension and receive the multiple files that are described in the package manifest, as taught by Rowley, where the list contains a hash for one or more of the files, and further creating a hash for the one or more received files, and comparing the created hash with the corresponding file hashes contained in the manifest to ascertain security of the file, as taught by Swank, since this allows for integrity checking of individual files that are delivered to the client.

In regard to Claims 49 and 50 it would be obvious to include the method of Claim 47 on a media and a computer system, the method of Claim 47 describes software and networks, which would inherently involve the need for the method to be implemented on a computer system, and stored on a computer media.

In regard to Claim 51, Rowley teaches receiving a package manifest containing a list of multiple files that comprise a software update of an older version of software (Column 1, lines 40-43) and comparing one or more files of an older version of a software extension with a newer version stored in a manifest, and if corresponding files are different, in that the manifest contains a file that does not contain a corresponding file on the client, the new file is downloaded from the web server. Rowley does not teach that the manifest contains hashes for one or more files

Art Unit: 2122

comprising the newer version of the software extension, and wherein comparing includes comparing hashes of different versions of files. Furthermore, Rowley does not teach that if the corresponding hashes are the same, then copying a file from an old local directory to a new local directory on the client associated with the newer version. Swank, however, does teach storing a file hash for an individual file to be updated on a computer system, creating an updated hash of a received file, and comparing the hashes to determine the integrity of the file (Column 3, lines 45-51). So that, when the hash of an old file, and a newly received file is the same, the old file is updated and replaced by the new file. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to receive a list of files of a new version of software, and compare the new files to old files on the client, and if the corresponding files are different, downloading a new file from the web server, as taught by Rowley, where the comparing is done by comparing hashes, and if the hashes are the same, updating an old version of a file with a new version, as taught by Swank, since hashes are effective ways of determining file integrity.

In regard to Claims 53 and 54 it would be obvious to include the method of Claim 47 on a media and a computer system, the method of Claim 47 describes software and networks, which would inherently involve the need for the method to be implemented on a computer system, and stored on a computer media.

11. Claims 25 and 29-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Swank (U.S. Patent Number 4,641,274) and further in view of Collins, III et al. (U.S. Patent Number 5,845,090).

In regard to Claim 25, Rowley and Swank teach the data structure of Claim 21, but do not teach that the file groups identify files that are to be downloaded before any other files. Collins,

Art Unit: 2122

however, does teach grouping files in data packages, and ordering the data packages on a queue for downloading (Column 5, lines 35-37). In this way, groups of files on a queue are prioritized over groups of files behind it. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 21, where the file groups identify files that are to be downloaded before any other files, since this allows critical files from being downloaded first.

In regard to Claim 29, Rowley and Swank teach the data structure of Claim 21, but do not teach that the file groups indicate file download priority. Collins, however, does teach grouping files in data packages, and ordering the data packages on a queue for downloading (Column 5, lines 35-37). In this way, groups of files on a queue are prioritized over groups of files behind it. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 21, where the file groups indicate file download priority, since this allows critical files from being downloaded first.

In regard to Claim 30, Rowley and Swank teach the data structure of Claim 21, and Rowley does not teach that one or more hashes are used for security. Swank, however, does teach using a hash for determining the integrity of a file (Column 3, lines 45-51). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 21, where the hashes are used for security, since this allows for a check that the file has not been tampered with when updating computer files on a computer system.

In regard to Claim 31, Rowley and Swank teach the data structure of Claim 21, and Rowley does not teach that one or more hashes are used for versioning. Swank, however, does

Art Unit: 2122

teach storing a file hash for an individual file to be updated on a computer system (Column 3, lines 45-51). A hash can identify individual files, and hence different versions of the same file would contain different hashes. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 21, where individual manifests can contain one or more file hashes used for versioning, as taught by Swank, since this allows different versions of the same file to be uniquely identified.

12. Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Swank (U.S. Patent Number 4,641,274) and further in view of Mastrianni et al. (U.S. Patent Number 6,615,276).

In regard to Claim 26, Rowley and Swank teach the data structure of Claim 21, but do not teach that the file groups identify files that are to be downloaded for offline use. Mastrianni, however, teaches a directory in a computer that collects web page files that are for offline use, and hence offline files are grouped in the directory (Column 8, lines 61-64). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 21, where the file groups identify files that are to be downloaded for offline use, as taught by Mastrianni, since this allows offline content to be accessible without need for going online.

13. Claim 33-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Swank (U.S. Patent Number 4,641,274) and further in view of Staelin (U.S. Patent Number 5,835,777).

In regard to Claim 33, Rowley and Swank teach the method of Claim 21, but do not teach including in the data structure one or more class identifiers for individual dynamic link libraries.

Art Unit: 2122

Staelin, however, does teach including an identifier for a shared library (Column 2, lines 57-60). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 21, where the data structure includes one or more class identifiers for individual dynamic link libraries, as taught by Staelin, since this allows new software to access its dependent libraries, without the need for additional downloads.

In regard to Claim 34, Rowley and Swank teach the method of Claim 21, but do not teach including in the data structure one or more DLL load dependencies. Staelin, however, does teach generating a dependency list of library load dependencies, and including these libraries in the manifest (Column 2, lines 51-60). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 21, where the data structure includes one or more DLL load dependencies, since this allows libraries dependent on a certain library to be downloaded with a library, thus preventing additional downloads, and ensuring a functional piece of software on delivery.

In regard to Claim 35, Claim 35 corresponds directly with Claims 32-34, and is rejected for the same reasons as Claims 32-34.

14. Claims 48 and 52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Swank (U.S. Patent Number 4,641,274) and further in view of Iannucci et al. (U.S. Patent Number 6,219,698).

In regard to Claims 48, Rowley and Swank teach the method of Claim 47, but do not teach the package manifest comprises an XML file. Innaucci, however, does teach a manifest file in the form of a web page, which is sent to the client for use in downloading a software extension (Column 5, lines 47-51). XML is a well-known and popular language for creating web pages.

Art Unit: 2122

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 47, where the package manifest comprises an XML file, as taught by Innaucci, since this allows the manifest file to take the form of a web page with XML features. Claim 52 corresponds directly with Claim 48 and is rejected for the same reasons as Claim 48.

15. Claims 58 and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Iannucci et al. (U.S. Patent Number 6,219,698) and further in view of “The Component Object Model: A Technical Overview”, by Sara Williams et al., Microsoft Corporation, October, 1994 (hereinafter Williams).

In regard to Claim 58, Rowley and Iannucci teach the data structure of Claim 55, but do not teach a third tag indicative of a COMClass ID. Williams, however, teaches the benefit of using COMClass ID's (Page 5, lines 42-53). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 55, where the data structure includes a tag indicative of a COMClass ID, whose benefit is taught by Williams, since this allows components to be uniquely named.

In regard to Claim 59, Rowley and Iannucci teach the data structure of Claim 55, and Rowley further teaches a third tag indicative of file load dependencies (Column 2, lines 55-58), but neither teach a fourth tag indicative of COMClass ID. Williams, however, teaches the benefit of using COMClass ID's (Page 5, lines 42-53). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 55, where the data structure includes a tag indicative of file load dependencies and a tag indicative of a

Art Unit: 2122

COMClass ID, whose benefit is taught by Williams, since this allows components to be uniquely named.

16. Claim 60 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Iannucci et al. (U.S. Patent Number 6,219,698) and further in view of Collins, III et al. (U.S. Patent Number 5,845,090).

In regard to Claim 60, Rowley and Iannucci teach the data structure of Claim 55, but do not teach that the ordering of the file groups implicitly defines a download order of the files. Collins, however, does teach grouping files in data packages, and ordering the data packages on a queue for downloading (Column 5, lines 35-37). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the data structure of Claim 55, where the ordering of the file groups implicitly defines a download order of the files, as taught by Collins, since this allows an order to be defined to groups of files, where some groups are more critical than others.

17. Claim 64, 65, 67, and 68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Collins, III et al. (U.S. Patent Number 5,845,090) in view of Carpenter et al. (U.S. Patent Number 5,859,973).

In regard to Claim 64, Collins teaches a download queue method that controls when files are to be downloaded to a client (Figure 2, item 20), the files pertaining to a software extension 25-30). Collins does not teach ascertaining whether a user action at the client requires one or more files that are not currently being downloaded, and manipulating the download queue responsive to a user action so that one or more required files are downloaded sooner than they would otherwise be. Carpenter, however, does teach queue manipulation based on user input

Art Unit: 2122

(Column 7, lines 21-25). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform a download queue method that controls when files are to be downloaded to a client, the files pertaining to a software extension, as taught by Collins, where the method further ascertains whether a user action at the client requires one or more files that are not currently being downloaded, and manipulating the download queue responsive to a user action so that one or more required files are downloaded sooner than they would otherwise be, since this allows a user to prioritize files that need to be downloaded while the queue is in progress.

In regard to Claim 65, Collins teaches a package object (Figure 3, item 39) with a list of files (Figure 3, item 36). Carpenter teaches queue manipulation by reordering objects, and has the functionality to move a package object to the head of the queue (Column 7, lines 21-25).

In regard to Claim 67, the examiner takes official notice that a file name is an inherent identifier of a file, and searching for a file name is a well-known method of locating a file.

In regard to Claim 68, the examiner takes official notice that a computer readable media must necessarily exist in order to perform the method of Claim 64, since the method of claim 64 is a computerized method.

18. Claim 66 rejected under 35 U.S.C. 103(a) as being unpatentable over Collins, III et al. (U.S. Patent Number 5,845,090) in view of Carpenter et al. (U.S. Patent Number 5,859,973) and further in view of Van Huben et al. (U.S. Patent Number 5,826,265).

In regard to Claim 66, Collins and Carpenter teach the method of Claim 64, but do not teach that said step of ascertaining comprises determining whether a required file is associated with a package object whose files are currently being downloaded. Van Huben, however, does

Art Unit: 2122

teach manipulating a queue so that files dependent on files currently being downloaded are moved to the head of the queue (Column 114, lines 65-67 and Column 115, lines 1-12).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 64, where said step of ascertaining comprises determining whether a required file is associated with a package object whose files are currently being downloaded, as taught by Van Huben, since this allows dependent files to be downloaded at the same time.

19. Claim 69 is rejected under 35 U.S.C. 103(a) as being unpatentable over Halpern et al. (U.S. Patent Number 6,282,711) in view of Taylor (U.S. Patent Number 5,721,824).

In regard to Claim 69, Halpern teaches: (a) identifying end user features (Figure 2, item 2B-1); (c) creating individual software packages for end user features (Figure 2, item 5); and (e) hosting the software packages on a web server (Column 1, lines 15-20). Halpern does not teach identifying shared dependencies between the end user features and creating individual software packages for the shared dependencies. Taylor, however, does teach identifying shared dependencies between the end user features (Figure 2A, item 100) and creating individual software packages for the shared dependencies (Figure 3, item 129). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to identify end user features, create individual software packages for end user features and host the software packages on a web server, as taught by Halpern, where shared dependencies are identified between the end user features and creating individual software packages for the shared dependencies, as taught by Taylor, since this allows software to function once it is installed without the need for additional installs.

20. Claim 70 is rejected under 35 U.S.C. 103(a) as being unpatentable over Halpern et al. (U.S. Patent Number 6,282,711) in view of Taylor (U.S. Patent Number 5,721,824) and further in view of Rowley (U.S. Patent Number 5,999,740).

In regard to Claim 70, Halpern and Taylor teach the method of Claim 69, but do not teach that the packages comprise one or more files that are associated with a single application program that provides multiple different functionalities, and packages extend the functionalities that are provided by the single application program. Rowley, however, does teach software extension packages that update applications, and hence extend the functionalities to include updated functionalities (Figure 9). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 69, where the packages comprise one or more files that are associated with a single application program that provides multiple different functionalities, and packages extend the functionalities that are provided by the single application program, as taught by Rowley, since this allows users to add functionalities to preexisting applications.

21. Claim 77-80 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Iannucci et al. (U.S. Patent Number 6,219,698) and further in view of Bailey (U.S. Patent Number 4,910,663).

In regard to Claim 77, Rowley and Iannucci teach the method of Claim 76, but do not teach that the statistics are ascertained from scenario runs. Bailey, however, does teach ascertaining statistics on program files by running a number of tests on the program (Column 1, lines 22-28 and Column 1, lines 46-50). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the software tool of Claim 76, where

Art Unit: 2122

the file usage statistics are ascertained from scenario runs, as taught by Bailey, since scenarios allow controlled testing of numerous areas of program functionality.

In regard to Claim 78, Rowley and Iannucci teach the method of Claim 77, but do not teach that the scenarios have individual priorities. Bailey, however, does teach ordering of test cases based on coverage. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the software tool of Claim 77, where the scenarios have individual priorities, as taught by Bailey, since this allows for a specific ordering of test cases based on specific factors.

In regard to Claim 79, Rowley and Iannucci teach the method of Claim 76, but do not teach that the statistics are ascertained from scenario runs that are collected by running logs on various scenarios. Bailey, however, does teach placing commands in source code in order to log statistical information during a test case run (Column 1, lines 29-45). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the software tool of Claim 76, where the file usage statistics are ascertained from scenario runs that are collected by running logs on various scenarios, as taught by Bailey, since this allows collecting of statistics during the running of the scenario.

In regard to Claim 80, Rowley and Iannucci teach the method of Claim 76, but do not teach that the statistics are ascertained from scenario runs that are collected by running logs on various scenarios. Bailey, however, does teach placing commands in source code in order to log statistical information during a test case run (Column 1, lines 29-45). The examiner takes official notice that scenarios are most likely defined in separate source code files, and checkpoints that separate one scenario from another are the individual scenario files. Therefore, it would have

Art Unit: 2122

been obvious to one of ordinary skill in the art at the time of the invention to build the software tool of Claim 76, where the file usage statistics are ascertained from scenario runs that are collected by running logs on various scenarios, as taught by Bailey, since this allows collecting of statistics during the running of the scenario, and where the scenarios have individual checkpoints that separate one scenario from another, which is obviously the individual scenario files.

22. Claim 81 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Iannucci et al. (U.S. Patent Number 6,219,698) and further in view of Kolawa et al. (U.S. Patent Number 5,761,408).

In regard to Claim 81, Rowley and Iannucci teach the software tool of Claim 76, but do not teach that the statistics are ascertained dynamically by building a knowledge base that describes tasks that users typically accomplish. Kolawa, however, does teach a test suite, which is a collection of tests that test the complete functionality of a software program, hence generating statistics for all functions that could be accomplished. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the software tool of Claim 76, where the statistics are ascertained dynamically by building a knowledge base that describes tasks that users typically accomplish, as taught by Kolawa. Since this allows for a comprehensive group of tests to be run based on common functionalities.

23. Claims 83 and 85 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740)

In regard to Claim 83, Rowley does not teach that the data structure is defined in XML. However the examiner takes official notice that XML is a well-known and popular language for

Art Unit: 2122

creating web pages, and it would be obvious to embody the data structure as an XML file, since this would make the manifest accessible as a web page.

In regard to Claim 85, it would be obvious to include the method of Claim 82 on a media since the method of Claim 82 describes creating a software package manifest, which would inherently involve the need for the method to be stored on a computer media.

24. Claim 84 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Bailey (U.S. Patent Number 4,910,663).

In regard to Claim 84, Rowley teaches the method of Claim 82, but does not teach that the statistics are ascertained from scenario runs. Bailey, however, does teach ascertaining statistics on program files by running a number of tests on the program (Column 1, lines 22-28 and Column 1, lines 46-50). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 82, where the file usage statistics are ascertained from scenario runs, as taught by Bailey, since scenarios allow controlled testing of numerous areas of program functionality.

25. Claim 86-90 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bailey (U.S. Patent Number 4,910,663) in view of Kolawa et al. (U.S. Patent Number 5,761,408) and further in view of Collins, III et al. (U.S. Patent Number 5,845,090).

In regard to Claim 86, Bailey teaches running a number of test scenarios on a file which represents a program (Column 1, lines 22-28 and Column 1, lines 46-50), and assigning a priority to these tests by ordering them based on coverage (Column 8, lines 21-27). Bailey does not explicitly teach that the scenarios describe ways in which a user interacts with an application program. Kolawa, however, does teach a test suite, which is a collection of tests that test the

Art Unit: 2122

complete functionality of a software program; hence describing ways a user can use an application. Neither Bailey nor Kolawa teach sorting multiple files in accordance with their scenario priority or priorities or downloading sorted files in an order defined by said sorting. Collins, however, does teach sorting data packages on a queue for downloading (Column 5, lines 35-37). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to run a number of test scenarios on a file, which represents a program and assign a priority to these tests by ordering them based on coverage, as taught by Bailey, where the scenarios describe ways in which a user interacts with an application program, as taught by Kolawa, since this allows a full range of testing functionalities of a program, and further sorting multiple files in accordance with their scenario priority or priorities and downloading sorted files in an order defined by said sorting, as taught by Collins, since this allows critical application to be downloaded before non-critical applications.

In regard to Claim 87, Collins teaches that the files on the queue are divided into packages, and hence the packages represent different file groups (Figure 3).

In regard to Claim 88, Bailey teaches sorting test scenarios in order of coverage, which is the percentage of instructions in each file that get used (Column 1, lines 22-28).

In regard to Claim 89, Claim 89 corresponds directly with Claims 87 and 88, and is rejected for the same reasons as Claims 87 and 88.

In regard to Claim 90, it would be obvious to include the method of Claim 86 on a media since the method of Claim 86 describes a software application program, which would inherently involve the need for the method to be stored on a computer media.

Art Unit: 2122

26. Claim 91, 92, 95, and 96 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rowley (U.S. Patent Number 5,999,740) in view of Collins, III et al. (U.S. Patent Number 5,845,090) and further in view of Bailey (U.S. Patent Number 4,910,663).

In regard to Claim 91, Rowley teaches sorting multiple files into multiple directories, where a directory is a file group (Column 2, lines 25-28). Rowley does not teach sorting multiple files based on scenario priority. Collins, however, does teach sorting data packages on a queue for downloading (Column 5, lines 35-37), and hence prioritizing certain packages. Neither Rowley nor Collins teaches sorting multiple files based on file usage order. Bailey, however, teaches sorting test scenarios in order of coverage, which is the percentage of instructions in each file that get used (Column 1, lines 22-28). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to sort multiple files into file groups as taught by Rowley, sort multiple files based on scenario priority, as taught by Collins, and sort multiple files based on file usage order, as taught by Bailey, so that files that get used the most, have priority over less used files.

In regard to Claim 92, Rowley teaches a manifest, but does not teach that the manifest is defined in XML. However the examiner takes official notice that XML is a well-known and popular language for creating web pages, and it would be obvious to embody the data structure as an XML file, since this would make the manifest accessible as a web page.

In regard to Claim 95, it would be obvious to include the method of Claim 91 on a media since the method of Claim 91 describes files to be downloaded a client, which would inherently involve the need for the method to be stored on a computer media.

Art Unit: 2122

In regard to Claim 96, it would be obvious to include the method of Claim 91 on a server computer since the method of Claim 91 describes files to be downloaded a client which would inherently involve the need for a server in a client/server environment.

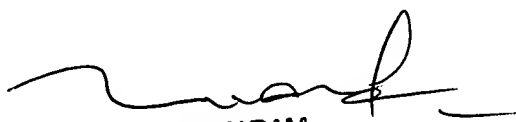
Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth A Gross whose telephone number is (703) 305-0542. The examiner can normally be reached on Mon-Fri 7:30-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

KAG


TUAN DAM
SUPERVISORY PATENT EXAMINER